LBL-35312
UC-600

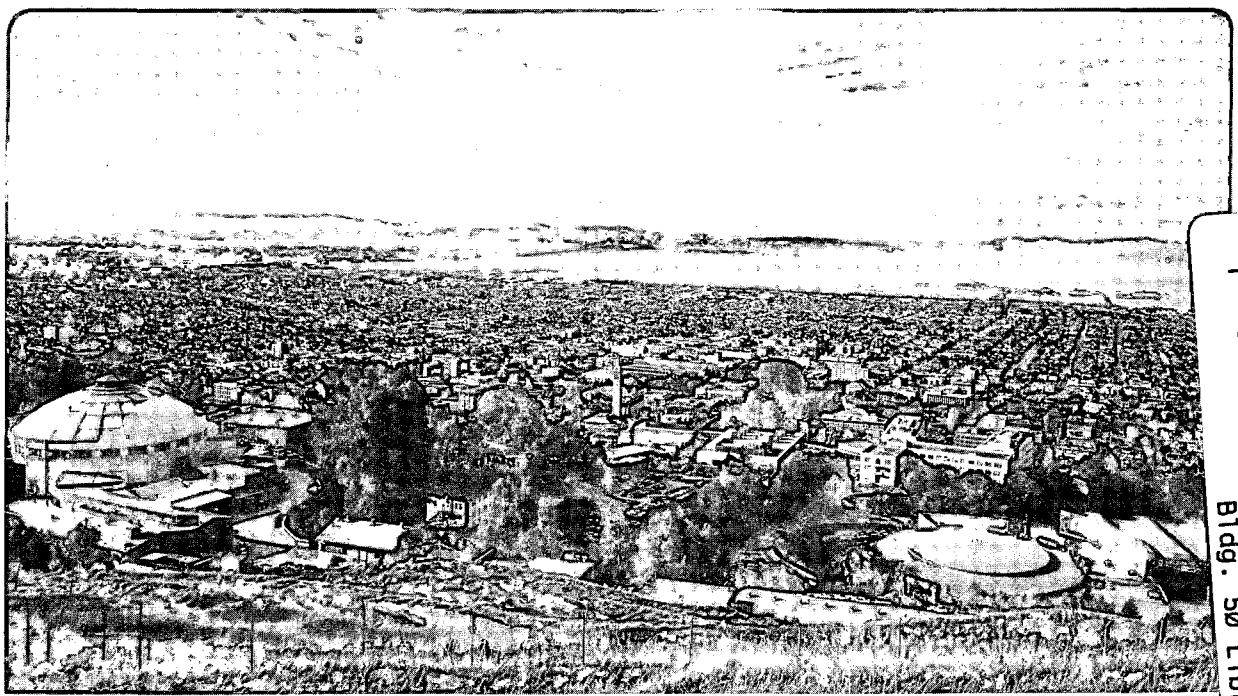
Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

EARTH SCIENCES DIVISION

A Code to Compute Borehole Temperature Profiles Under Steady Flow Conditions

F.V. Hale and C.F. Tsang

February 1994



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

DISCLAIMER

LBL-35312
UC-600

A Code to Compute Borehole Temperature Profiles Under Steady Flow Conditions

F.V. Hale and C.F. Tsang

Earth Sciences Division
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

February 1994

This work was carried out under U.S. Department of Energy Contract No. DE-AC03-76SF00098 for the Director, Office of Civilian Radioactive Waste Management, Office of External Relations, administered by the Nevada Operations Office in cooperation with the Swiss National Cooperative for Disposal of Radioactive Waste (Nagra).

Table of Contents

1. INTRODUCTION

If an uncased section of a wellbore intersects a number of flowing fractures or permeable layers containing fluids with different temperatures, it may be possible to use temperature logs of the wellbore fluid in order to obtain information about the hydraulic properties of the geological formation. One possible method is to first flush the wellbore with water of a known temperature and then pump the well at a small, constant flow rate. During pumping a number of logging runs are made, and distinctive features in the logs develop as fluid from each inflow point enters the wellbore.

Section 2 of this report describes the analytical and numerical solutions to the governing equations. Section 3 presents a brief discussion of the FORTRAN code Boretemp which incorporates the numerical solution. Section 4 describes the input and output files for the code, and Section 5 contains verification cases and examples.

2. ANALYTICAL AND NUMERICAL SOLUTIONS

2.1 Governing Equations

The borehole is discretized into a number of uniform, cylindrical cells for flow and convection-conduction calculations down the borehole. Surrounding each borehole cell is a set of concentric rock rings for radial heat conduction calculations.

The origin of the coordinate system is at elevation zero at the center of the borehole. The x axis increases down the center line of the borehole, and the y axis increases radially from the center of the borehole.

The numerical equations for heat transport are given below, and the major symbols are defined in Table 1. The subscripts i and j refer to the borehole cell or rock layer (axial index) and rock ring (radial index), respectively. Equation (1) is for heat transport in the wellbore and across the wellbore/rock interface, and Equation (2) is for radial heat transport within the a rock layer.

$$\frac{\Delta E_i}{\Delta t} = \frac{1}{V_w}\left[F_i + \left(E_{i-1}\, q_{i-1/2} - E_i\, q_{i+1/2}\right) + \right. \tag{1}$$

$$\lambda_w \frac{A_w}{\Delta x}\frac{1}{\rho_w c_w}\left(E_{i+1} - 2E_i + E_{i-1}\right) +$$

$$\left. \lambda_r\, 2\pi\Delta x\, \frac{R_w}{R_1 - R_w}\left(\frac{E_{i1}}{\rho_r c_r} - \frac{E_i}{\rho_w c_w}\right)\right]$$

$$\frac{\Delta E_{ij}}{\Delta t} = \frac{1}{V_j} \lambda_r \, 2\pi\Delta x \, \frac{1}{\rho_r c_r} \left[\frac{R_{(j-1/2)}}{R_j - R_{j-1}} \left(E_{i(j-1)} - E_{ij}\right) - \frac{R_{(j+1/2)}}{R_{j+1} - R_j} \left(E_{ij} - E_{i(j+1)}\right) \right] \quad (2)$$

The terms of Equation (1) within the square braces represent the effects of (a) inflows into the borehole, (b) axial advection within the borehole, (c) axial conduction within the borehole, and (d) radial conduction between the borehole and the rock.

Heat transport within the rock mass is only by radial conduction, and the outside boundary of the rock cylinder is closed. There is no conduction between rock layers.

The radial distance used for the numerical calculation is from the center of one ring to the center of the next, except for the center ring which is adjacent to the borehole.

Table 1. Definition of principal symbols.

A	wellbore cross-sectional area	m^2
c	heat capacity	J / kg-K
E	energy density (1D in wellbore, 2D in rock)	J / m^3
F	energy source (inflow)	J / sec
λ	thermal conductivity	J / sec-m-K
q	volumetric flow rate	m^3 / sec
R	radial distance to node centers; increasing outward; origin at center of wellbore	m
R_w	wellbore radius	m
r	subscript referring to rock	
ρ	density	kg / m^3
t	time	sec
V	volume of rock ring	m^3
V_w	volume of wellbore section	m^3
w	subscript referring to water	
x	depth; increasing downward; origin at surface	m

The energy density, E, can be converted to temperature, T, in degrees Celsius by the following equation:

$$T = \frac{E}{\rho c} - 273.15$$

Thermodynamic properties for the borehole fluid and the rock mass, initial energy values and problem geometry are specified in the input.

2.2 Discretization in Space

The geometry of the rock rings depends on three input values: (1) the borehole segment length (Δx), which becomes the thickness of the rock layer; (2) the borehole diameter (d), which determines the inside radius of the first ring; and (3) the specified boundary radius at which rings change from having a uniform, small thickness to an exponentially increasing thickness (R_c).

Rock rings in the region $R_w < R < R_c$ will have uniform thickness of one-eighth of the well diameter (d/8). Beyond R_1, rock rings have a thickness which increases by a factor of 1.2 with each ring (i.e. 0.15d, 0.18d, 0.216d, 0.259d, etc.) until an outside radius of 10 times R_c is reached. This scheme is intended to provide a sufficient volume of rock so that the temperature in the cells of the outside ring (with a closed boundary) will not change by more than one degree Kelvin during the run. If a temperature change of more than one degree Kelvin is detected, the program stops and instructs the user to specify a larger rock mass.

For example, if d is 10 cm and R_c is 1 m, there will be 76 rock layers, 0.5 m < y < 1 m, with thickness 0.0125 m, then 28 layers of increasing thickness, the outermost layer being 1.717 m thick and extending from 9.5 m to 11.2 m.

2.3 Discretization in Time

If at any time the code attempts to transport more than half of the energy in a cell, the program stops and instructs the user to start again with a smaller maximum time step. In general, the code uses the maximum time step specified in the program description input file (unless advection requires a still smaller step).

2.4 Calculation of Flow Rates

Flow rates are assumed to be constant with time, i.e. a steady state flow is assumed. The flow rates into the bottom and out of the top of the region, as well as the flow rate for each of the feed points are included in the problem description.

2.5 Calculations at Each Time Step

Conservation of energy is verified during each time step by comparing the total energy in the system at the end of the time step with the expected value based on the initial conditions, inflows and outflows. A conservation of energy error is indicated (and the run stopped) if the difference between the actual and expected values is greater than 0.1% of the total input energy.

3. DESCRIPTION OF FORTRAN CODE BORETEMP

The program consists of a main routine, three subroutines and one function. The main program controls the size of arrays and calls two subroutines: (1) **rdprob** to read in the problem description file and set up the initial conditions, and (2) **tsteps** to perform the calculation. The function **iconvx** is used by subroutine rdprob to convert depths in meters to cell indices. The subroutine **cprt** is called by subroutine tsteps to produce temperature profile output.

There are no common blocks used in the code; all data is passed between routines by parameter lists.

Array dimensions are controlled by a group of parameter statements at the beginning of the main program. These define the program size limits for the number of borehole cells, number of rock rings, number of inflows, and number of output logs.

4. INPUT AND OUTPUT GUIDE

4.1 Input File

The model uses one input and two output files. The input file contains the problem description consisting of borehole geometry, top and bottom borehole flows, feed point flows, timing parameters, dispersion parameters, and initial values. One of the output files consists of messages produced by the model, while the other contains temperature profiles at the requested output times.

Table 2 provides an annotated example of the problem input file, and Table 3 defines the variable names, units and alternative inputs.

4.2 Output Files

The program produces two output files. The first contains various informational messages, and the second contains temperature profiles at the initial time and at the times specified in the problem description. Three sets of temperature profiles are produced at each output time: (1) the wellbore profile, (2) the profile in the rock at radius R_c, the limit of the rings of uniform thickness, and (3) the profile in the outside rock ring, which should be constant.

Table 2. Annotated sample input file.

0 100 1 10	Borehole geometry. Borehole sections from 0 m to 100 m depth at 1 m intervals, with a diameter of 10 cm.
1 850. 2.00 2.7e3	Rock geometry and thermal properties. Radius for small rock rings is 1 m from center of well, implying outside radius for rock is 10 m; rock heat capacity is 850 J/kg-K; rock thermal conductivity is 2.00 J/s-m-K; and rock density is 2,700 kg/m^3.
0 0.4	Borehole inflow at bottom (0 l/min) and outflow at top (0.4 l/min)
3 4178.2 0.607 1.e3	Borehole fluid properties: 3 inflows follow; fluid heat capacity is 4178.2 J/kg-K; fluid thermal conductivity is 0.607 J/s-m-K; and fluid density is 1,000 kg/m^3.
25 0.1 8.5 0.	First inflow, at depth 25 m with flow rate 0.1 l/min and temperature 8.5 C starting at time 0 h.
62 0.2 12.2 0.	Second inflow.
80 0.1 14. 0.	Third inflow
0 100 1 4	Problem timing parameters. Start at 0 h, run to 100 h, time step of 1 min; output temperature profile at four times specified below.
0.0833333	Output temperature profile at time of 5 min.
1.0	Output temperature profile at 1 hr.
10.0	Output temperature profile at 10 hrs
100.0	Output temperature profile at 100 hrs
-1	Indicate that a constant initial borehole temperature follows.
20.	Constant initial temperature in borehole is 20° C.
0. 20.	Constant initial temperature of rock is 20° C, with no thermal gradient (0° C/m).

Table 3. Input file definition.

Read List	Description	Units
xtop,xbot,delx,diam	xtop: depth of top of modeling section	m
	xbot: depth of bottom of modeling section	m
	delx: length of uniform borehole section (rock layer thickness)	m
	diam: borehole diameter (converted internally to meters)	cm
rad1,ecapr,econr,denr	rad1: radial extent of uniform thickness rings	m
	ecapr: heat capacity of rock	J / kg-K
	econr: thermal conductivity of rock	J / s-m-K
	denr: density of rock	kg / m^3
qexti,qexto	qexti: flow into bottom of section (converted internally to m^3/sec)	l / min
	qexto: flow out of top (recomputed if necessary; converted internally to m^3/sec)	l / min
iflim,ecapw,econw,denw	iflim: number of inflows	
	ecapw: heat capacity of borehole fluid	J / kg-K
	econw: thermal conductivity of borehole fluid	J / s-m-K
	denw: density of borehole fluid	kg / m^3
xin(i),qin(i),ein(i),t0(i)	xin: depth of inflow	m
(i=1,iflim)	qin: inflow rate (internally converted to m^3/sec)	l / min
Multiple lines	ein: inflow temperature (internally converted to energy density, J / m^3)	C
	t0: time at which inflow begins flowing (internally converted to sec)	hr
tstart,tend,delt,ilpt	tstart: starting time for run (internally converted to sec)	hr
	tend: ending time for run (internally converted to sec)	hr
	delt: maximum time step (internally converted to sec)	min
	ilpt: number of output logs requested	
pt(i)	pt: output log time (internally converted to sec)	hr
(i=1,ilpt) *Multiple lines*		
ie0n	ie0n: initial borehole temperature indicator if >0, number of initial values following if <0, a single initial temperature for borehole fluid follows	
x(i),e0(i)	x: depth for initial temperature	m
(i=1,ie0n if ie0n>0)	e0: initial temperature (internally converted to energy density, J / m^3)	C
Alternate multiple lines		
e0	e0: initial uniform temperature for borehole fluid (internally converted to energy density, J / m^3)	C
Alternate line if ie0n<0		
arock,brock	arock: slope for rock temperature gradient	K / m
	brock: constant for rock temperature gradient (internally converted to K) (T = x * arock + brock)	C

5. CODE VERIFICATION

Five runs of the code are discussed in this section; the first three provide comparisons between code results and analytical solutions, and the last two are additional examples.

5.1 Comparison with Analytical Solution for Transient Radial Conduction

An analytical solution is available for transient heat conduction through a solid ring with constant temperature at the inside and outside radii (Carslaw and Jaeger, "The Conduction of Heat in Solids," 2nd ed., Oxford, 1959, p. 337). Boretemp was run with a borehole radius (inside radius of conducting cylinder) of one meter and borehole fluid having a fixed temperature of 100° C. The initial temperature of the surrounding rock (the conducting cylinder) was 0° C. Rock at the outside radius of 100 meters was kept at a constant temperature of 0° C. In order to accommodate the exponential time scale in the published solution, three Boretemp runs were made with different time steps.

In order to obtain a constant inside temperature, the "fluid" in the borehole had an artificial heat capacity, and inflows were placed in every cell. Cells were spaced one meter apart. Identical results were obtained for each cell layer.

The general form of the input file can be seen by the input for the middle range of times:

```
0 10 1 200
10 800 3 2500          r1=10m, rmax=100m, typical geol material
0 100
10 1.e9 0.607 1000     special high heat-cap water for const. temp.
 0.9 10 100 0
 1.9 10 100 0
 2.9 10 100 0
 3.9 10 100 0
 4.9 10 100 0
 5.9 10 100 0
 6.9 10 100 0
 7.9 10 100 0
 8.9 10 100 0
 9.9 10 100 0
0 185185 15 10
5.556
18.52
55.56
185.2
555.6
1852
5556
18519
55556
185185
-1
100.
0. 0.
```

Line two of the input gives the thermodynamic properties of the generic geological material:

Heat capacity, c_r	800 J / kg-K
Thermal conductivity, λ_r	3 J / sec-m-K
Density, ρ_r	2,500 kg / m^3

This gives a thermal diffusivity, κ_r, of

$$\kappa_r = \lambda_r / \rho_r c_r = 1.5 \times 10^{-6} \text{ m}^2/\text{sec}$$

Since R_w, the inside radius of the cylinder is one meter, κ_r/R_w^2 is 1.5×10^{-6} sec^{-1}.

The different time steps used to compute the different curves were:

Δt	$\kappa_r t / R_w^2$
15 sec	0, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3
15 min	1, 3, 10, 30, 100
2 hr	300, 1000, 3000, 10000, 30000, 100000

Figure 1 shows the comparison between the numerical results (circles) for $\kappa_r t / R_w^2$ ranging from 0.03 to 1000 with values obtained from Figure 41 of Carslaw and Jaeger (lines). The agreement between the analytical and numerical solutions in excellent.

5.2 Comparison with Analytical Solution for Steady State Radial Conduction

The steady state solution for radial conduction through a solid with constant temperature boundaries at inside and outside radii is a logarithmic temperature distribution. By running the above case to a much longer time, Boretemp was used to compute the steady state radial conduction in a cylinder with a fixed temperature at inside and outside radii. The inside radius was one meter with a fixed temperature of 100° C, and the outside radius was 100 m with a fixed temperature of 0° C. The initial temperature of the rock cylinder was 0° C. The numerical result shown in Figure 2 is for $\kappa_r t / R_w^2$=10,000 (circles). The solid line is the analytic solution, a logarithmic temperature drop with increasing radius.

5.3 Comparison with Analytical Solution for Single Source, Upflow, with Radial Conduction

A Laplace-domain analytical solution for heat migrating along a borehole from a single source, with radial conduction into the rock, was developed by D. Guyonnet (private communication, October 1993) and used by him to verify results from this code. A single source is used because, as Guyonnet reported, "no analytical solutions exist for the case of multiple inflows and therefore a simpler case must be considered....Solutions for the cases of closed and infinite-extent boundaries [exist]. As the latter is simpler to implement, it was selected and care was taken to ensure that the Boretemp simulations were not influenced by the outer boundary (i.e. the boundary was not reached by the heat pulse in the course of the simulation.)"

The problem description file for this case is specified below:

```
100 200 1 10              Section from depth 100m to 200m in 1m steps,
                          10cm diameter
1 850. 8.00 2.7e3         Uniform rock rings to 1m, typical rock
                          properties
0 10.                     Outflow of 10 l/min
1 4178.2 0.0607 1.e3      One inflow; typical water properties (except
                          conductivity)
 174.5 10. 30. 0.         Inflow at 174.5m, 10l/min, 30° C, starting at time
                               0hr
0 48 .1 4                 Run from 0 to 48h with 0.1m steps, 4 output
                          times.
6                         Outputs at 6, 12, 24 and 48 hours
12
24
48
-1                        Uniform initial well temperature of 20° C
20.
0. 20.                    Initial rock temperature of 20° C, no gradient.
```

The message output produced by the run is given below:

```
 Borehole segments:         100
 Segment length (m):   1.0000000000000000
  Each region has diameter          (m)    0.100000000000000006
              cross-sectional area (m**2) 0.78539816350000017OE-02
          and volume               (m**3) 0.78539816350000017OE-02

 Rock ring, min & max radius (m), vol (m**3)
  1     .0500     .0625          .00442
  2     .0625     .0750          .00540
  3     .0750     .0875          .00638
  4     .0875     .1000          .00736
  5     .1000     .1125          .00834
  6     .1125     .1250          .00933
  7     .1250     .1375          .01031
  8     .1375     .1500          .01129
  9     .1500     .1625          .01227
 10     .1625     .1750          .01325
 11     .1750     .1875          .01424
```

```
      ... additional rings of uniform thickness omitted...

 70     .9125     .9250          .07216
 71     .9250     .9375          .07314
 72     .9375     .9500          .07412
 73     .9500     .9625          .07510
 74     .9625     .9750          .07609
 75     .9750     .9875          .07707
 76     .9875    1.0000          .07805
 77    1.0000    1.0125          .07903
 78    1.0125    1.0275          .09613
 79    1.0275    1.0455          .11723
 80    1.0455    1.0671          .14336
 81    1.0671    1.0930          .17590
 82    1.0930    1.1241          .21665
 83    1.1241    1.1614          .26800
 84    1.1614    1.2062          .33316
 85    1.2062    1.2600          .41643
 86    1.2600    1.3245          .52368
 87    1.3245    1.4019          .66291
 88    1.4019    1.4948          .84518
 89    1.4948    1.6062         1.08575
 90    1.6062    1.7399         1.40593
 91    1.7399    1.9004         1.83546
 92    1.9004    2.0930         2.41618
 93    2.0930    2.3241         3.20703
 94    2.3241    2.6015         4.29140
 95    2.6015    2.9342         5.78756
 96    2.9342    3.3336         7.86361
 97    3.3336    3.8128        10.75904
 98    3.8128    4.3879        14.81553
 99    4.3879    5.0780        20.52139
100    5.0780    5.9061        28.57522
101    5.9061    6.8998        39.97763
102    6.8998    8.0922        56.16296
103    8.0922    9.5232        79.18888
104    9.5232   11.2403       112.00904
 Maximum time step (min) is          0.392699081750000067
     (based on advection only)
 Maximum linear velocity (m/sec) is  0.212206590761485313E-01
 Maximum Reynold's number is         7073.55302538284468
 qtmax =  0.166666666666666661E-03
 Number of time steps:                    28800
 Time step (min):                    0.100000000000000006
 Problem size (segments*time steps):  299520000
 Initial heat in borehole (J):            961986560.364570498
 Initial heat in rock (J):                26703550440499.4961
 Initial heat in system (J):              26704512427059.8594


 Printing profile at time (hrs) 0.00000000000000000E+00
 Printing profile at time (hrs)  6.00000000000000000
 Printing profile at time (hrs)  12.0000000000000000
 Printing profile at time (hrs)  24.0000000000000000
 Printing profile at time (hrs)  48.0000000000000000


 Final heat in borehole (J):              973858513.034498692
 Change in heat in borehole (J):          11871952.6712174416
 Final heat in rock (J):                  26704582586914.0820
 Change in heat in rock (J):              1032146414.58593750
```

```
Final heat in whole system (J):          26705556445427.1172
Change in heat in whole system (J):      1044018367.25781250
Heat added to system (J):                36478694304.0269470
Heat removed from system (J):            35434675935.1594238


*** PROBLEM FINISHED ***
```

Figure 3 shows the temperature profiles at 6, 12, 24 and 48 hours for a single source at depth 174.5m flowing into a 10 cm diameter borehole. The upper set is the borehole temperature and the lower set is the rock temperature at radius R_C (1 m). The borehole was flushed at time zero with fluid having a uniform temperature of 20° C, and the outside rock ring had a uniform temperature of 20° C throughout the run. Typical thermodynamic values for geological materials and water were used, and the pumping rate was 10 l/min.

Based on his analytical calculation, D. Guyonnet concluded that "the results of the [Laplace-domain] analytical solution (without longitudinal dispersion i.e., water thermal diffusivity), using the same parameters as for BORETEMP, agree well with those of the numerical code." The small circles on the six-hour temperature profile are from Guyonnet's analytical solution.

5.4 Additional Example: Constant Input Under Constant Overall Flow

This sample run is similar to the last case, except that there is an additional 10 l/min flow from a point lower in the wellbore. The result is that the temperature increase in the wellbore is more widely spread due to increased advection. The input file is shown below:

```
100 200 1 10
1 850. 8.00 2.7e3
0 20.                              Flow out is now 20 l/min
2 4178.2 0.607 1.e3                Now two inflows
 174.5 10. 30. 0.
 190.5 10. 20. 50.                 2nd inflow at 190.5 uses ambient temperature
                                         (starts late)
0 48 .1 4
6
12
24
48
-1
20.
0. 20.
```

The message output can be compared with that in section 5.1.

```
Borehole segments:          100
Segment length (m):    1.00000000000000000
 Each region has diameter          (m)     0.100000000000000006
            cross-sectional area (m**2) 0.785398163500000170E-02
         and volume               (m**3) 0.785398163500000170E-02

Rock ring, min & max radius (m), vol (m**3)
 1     .0500     .0625          .00442
 2     .0625     .0750          .00540
```

```
  3     .0750     .0875          .00638
  4     .0875     .1000          .00736
  5     .1000     .1125          .00834
  6     .1125     .1250          .00933
  7     .1250     .1375          .01031
  8     .1375     .1500          .01129
  9     .1500     .1625          .01227
 10     .1625     .1750          .01325
 11     .1750     .1875          .01424

       ... additional rings of uniform thickness omitted...

 70     .9125     .9250         .07216
  71     .9250     .9375          .07314
  72     .9375     .9500          .07412
  73     .9500     .9625          .07510
  74     .9625     .9750          .07609
  75     .9750     .9875          .07707
  76     .9875    1.0000          .07805
  77    1.0000    1.0125          .07903
  78    1.0125    1.0275          .09613
  79    1.0275    1.0455          .11723
  80    1.0455    1.0671          .14336
  81    1.0671    1.0930          .17590
  82    1.0930    1.1241          .21665
  83    1.1241    1.1614          .26800
  84    1.1614    1.2062          .33316
  85    1.2062    1.2600          .41643
  86    1.2600    1.3245          .52368
  87    1.3245    1.4019          .66291
  88    1.4019    1.4948          .84518
  89    1.4948    1.6062         1.08575
  90    1.6062    1.7399         1.40593
  91    1.7399    1.9004         1.83546
  92    1.9004    2.0930         2.41618
  93    2.0930    2.3241         3.20703
  94    2.3241    2.6015         4.29140
  95    2.6015    2.9342         5.78756
  96    2.9342    3.3336         7.86361
  97    3.3336    3.8128        10.75904
  98    3.8128    4.3879        14.81553
  99    4.3879    5.0780        20.52139
 100    5.0780    5.9061        28.57522
 101    5.9061    6.8998        39.97763
 102    6.8998    8.0922        56.16296
 103    8.0922    9.5232        79.18888
 104    9.5232   11.2403       112.00904
 Maximum time step (min) is          0.196349540875000034
     (based on advection only)
 Maximum linear velocity (m/sec) is  0.424413181522970626E-01
 Maximum Reynold's number is         14147.1060507656894
 qtmax =  0.333333333333333322E-03
 Number of time steps:                    28800
 Time step (min):                    0.100000000000000006
 Problem size (segments*time steps):  299520000
 NOTE--fracture          2
    t0 outside of problem time limits:   50.0000000000000000
 Initial heat in borehole (J):            961986560.364570498
 Initial heat in rock (J):                26703550440499.4961
 Initial heat in system (J):              26704512427059.8594
```

```
 Printing profile at time (hrs) 0.00000000000000000E+00
 Printing profile at time (hrs)  6.00000000000000000
 Printing profile at time (hrs)  12.0000000000000000
 Printing profile at time (hrs)  24.0000000000000000
 Printing profile at time (hrs)  48.0000000000000000


 Final heat in borehole (J):              970318864.533721566
 Change in heat in borehole (J):          8332304.17044031620
 Final heat in rock (J):                  26704304961709.6289
 Change in heat in rock (J):              754521210.132812500
 Final heat in whole system (J):          26705275280574.1641
 Change in heat in whole system (J):      762853514.304687500
 Heat added to system (J):                71754067008.0792389
 Heat removed from system (J):            70991213492.3437958


 *** PROBLEM FINISHED ***
```

Figure 4 shows the wellbore (upper) and rock (lower) profiles at 6 and 48 hours. Comparing this result with Figure 3 shows the effect of the regional flow advection.

5.5 Additional Example: Two Inflows and Temperature Gradient

One additional example is provided which combines multiple inflows with a downhole temperature gradient. The problem description file is:

```
100 200 1 10
1 850. 8.00 2.7e3
0 10.
2 4178.2 0.607 1.e3
 124.5  .5 17. 0.                Inflow at 124.5m, cool
 174.5  .5 25. 0.                Inflow at 175.5m, warm
0 10. .1 4
.1
.5
5
10
-1                               Constant borehole initial temp follows
20.
0.02 18.                         Gradient in rock is 0.02 C/m
```

Part of the message output is shown below:

```
 Borehole segments:         100
 Segment length (m):   1.00000000000000000
  Each region has diameter          (m)    0.100000000000000006
              cross-sectional area (m**2) 0.785398163500000170E-02
          and volume               (m**3) 0.785398163500000170E-02

 Rock ring, min & max radius (m), vol (m**3)
  1     .0500     .0625          .00442
  2     .0625     .0750          .00540
```

```
  3     .0750     .0875          .00638
  4     .0875     .1000          .00736
  5     .1000     .1125          .00834
...
      ... additional rings omitted...


100    5.0780    5.9061        28.57522
101    5.9061    6.8998        39.97763
102    6.8998    8.0922        56.16296
103    8.0922    9.5232        79.18888
104    9.5232   11.2403       112.00904
 Outflow at top changed by  0.14999999999999987E-03
                       to  0.16666666666666675E-04
 Maximum time step (min) is           3.92699081750000056
    (based on advection only)
 Maximum linear velocity (m/sec) is  0.212206590761485330E-02
 Maximum Reynold's number is         707.355302538284491
 qtmax =  0.16666666666666675E-04
 Number of time steps:                     6000
 Time step (min):                    0.100000000000000006
 Problem size (segments*time steps):   62400000
 Initial heat in borehole (J):            961986560.364570498
 Initial heat in rock (J):                26794642203898.7812
 Initial heat in system (J):              26795604190459.1445

 Printing profile at time (hrs) 0.00000000000000000E+00
 Printing profile at time (hrs) 0.100000000000000006
 Printing profile at time (hrs) 0.500000000000000000
 Printing profile at time (hrs)  5.00000000000000000
 Printing profile at time (hrs)  10.0000000000000000

 Final heat in borehole (J):              965337481.366964698
 Change in heat in borehole (J):          3350921.00368344784
 Final heat in rock (J):                  26794641214005.6992
 Change in heat in rock (J):             -989893.082031250000
 Final heat in whole system (J):          26795606551487.0664
 Change in heat in whole system (J):      2361027.92187500000
 Heat added to system (J):                737410518.000067115
 Heat removed from system (J):            735049490.179382682

 *** PROBLEM FINISHED ***
```

Figure 5 shows the curves for times 0, 0.1 and 10 hours for multiple inflows with a downhole temperature gradient. The borehole was originally flushed with water having a constant temperature of 20° C. Over time, the fluid temperature moves toward thermal equilibrium with the rock, and the different inflow signatures can be seen indicating inflow temperature above and below the gradient.

6. ACKNOWLEDGMENTS

Discussions and cooperation with NAGRA personnel, especially S. Vomvoris, P. Bluemling are much appreciated. We would also like to thank, D. Guyonnet of COLENCO and Christine Doughty of LBL for reviewing and commenting on the manuscript and software. This work was carried out under U.S. Department of Energy Contract No. DE-AC03-76SF00098 for the Director, Office of Civilian Radioactive Waste Management, Office of External Relations, administered by the Nevada Operations Office in cooperation with the Swiss National Cooperative for Disposal of Radioactive Waste (Nagra).

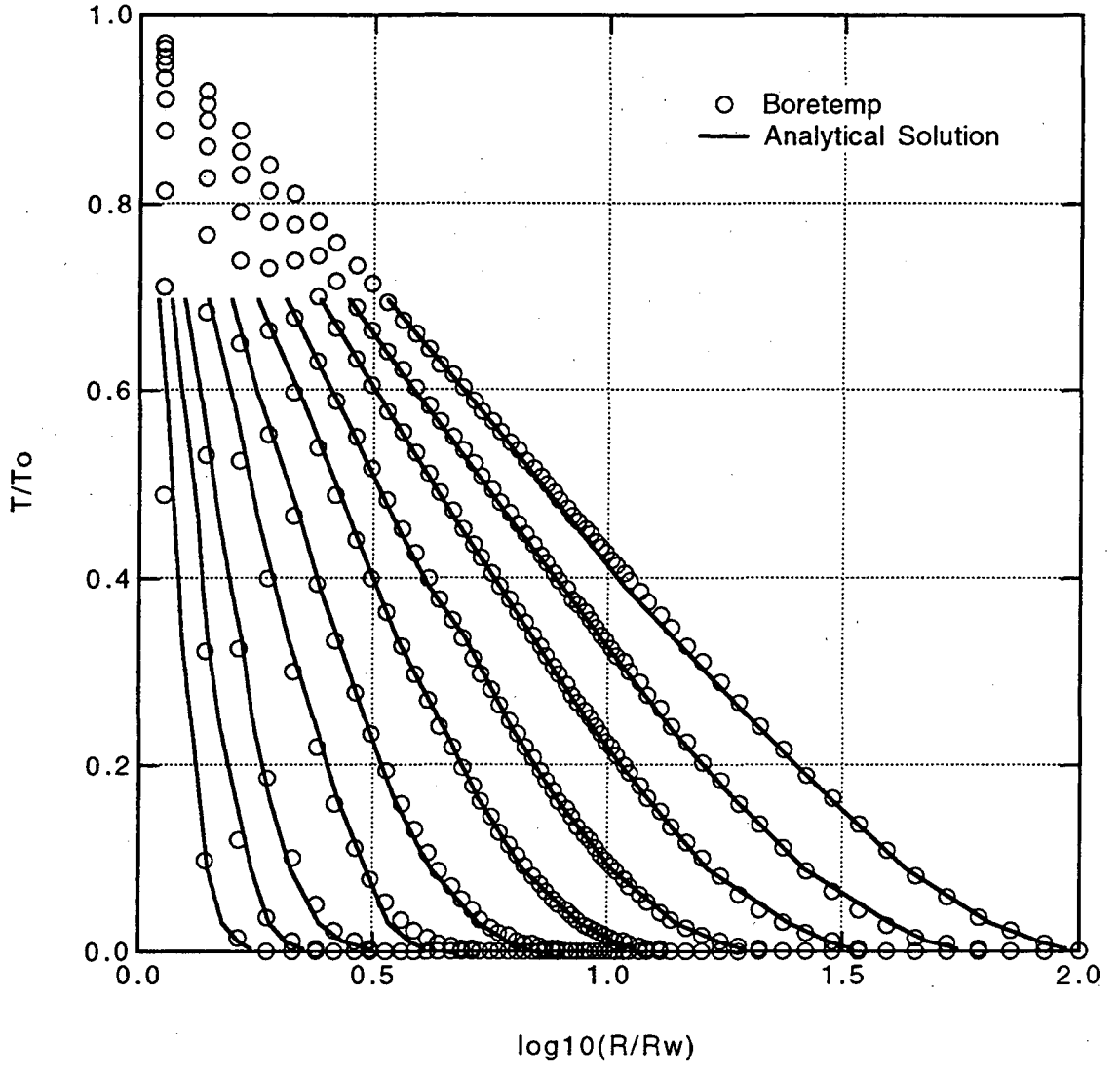


Figure 1. Comparison of Boretemp results with analytical transient solution for constant inside temperature, zero initial temperature cylinder

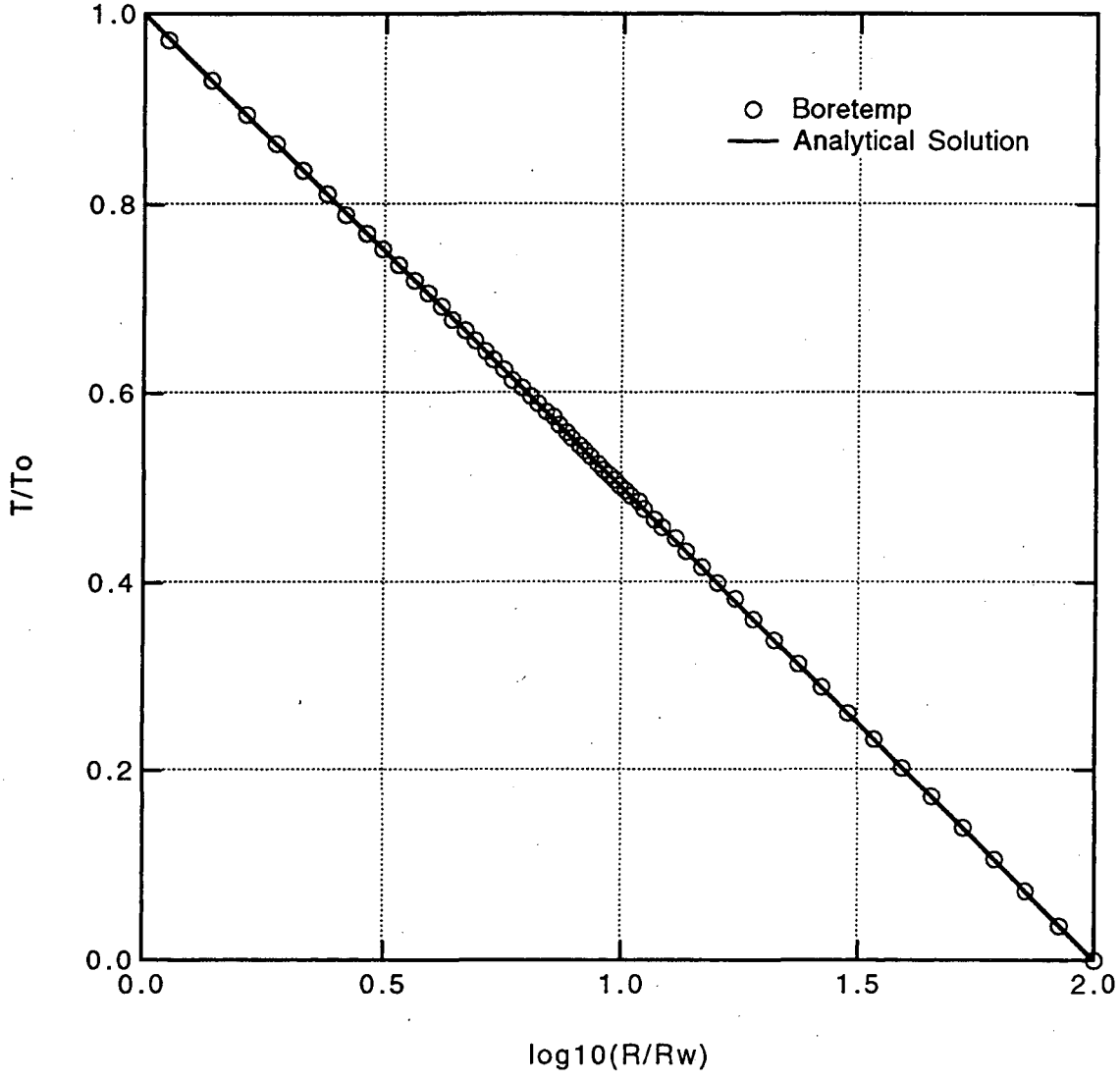


Figure 2. Comparison of Boretemp results with analytical steady state solution for constant inside and outside temperature cylinder .

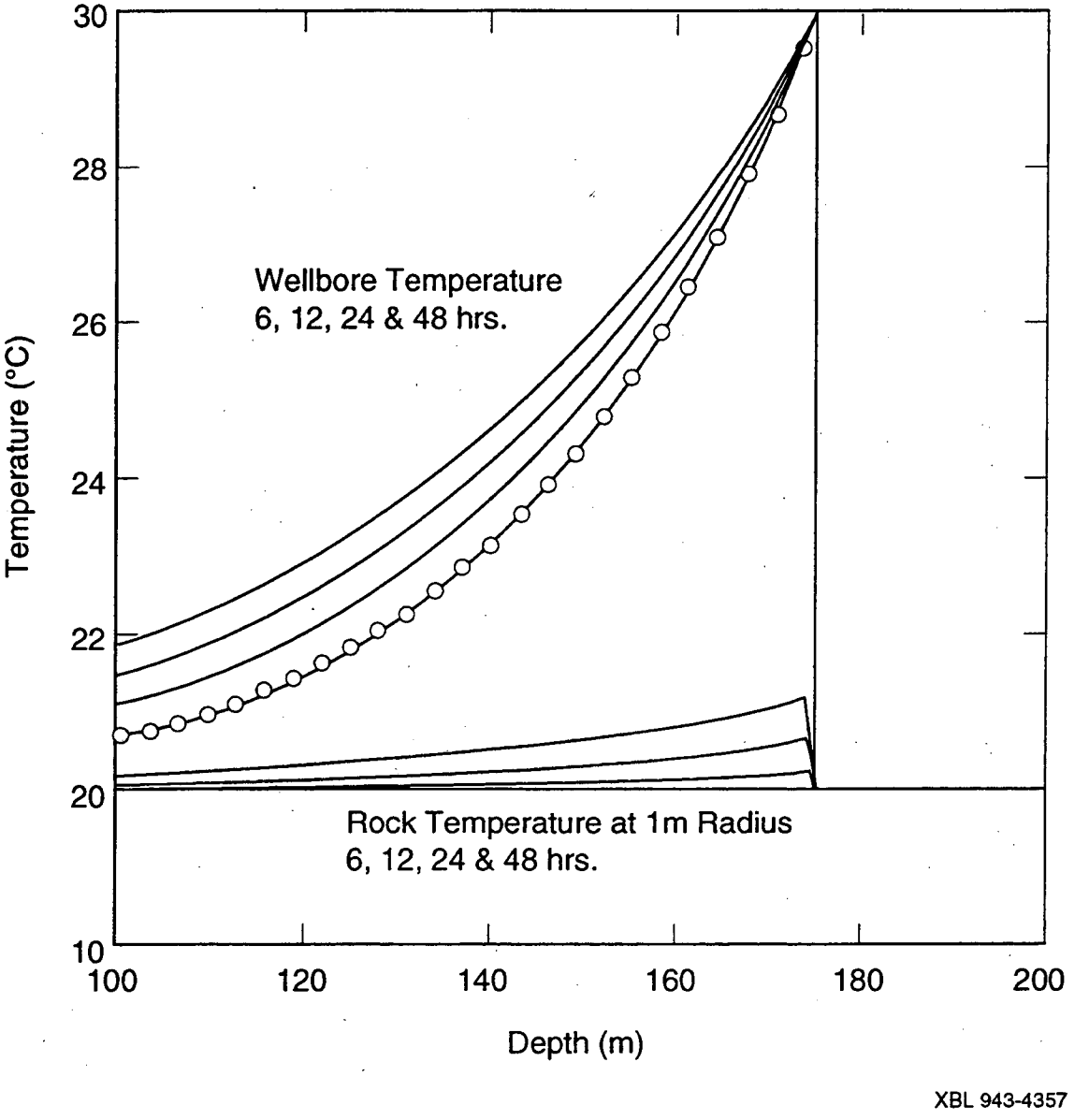


Figure 3. Numerical results for single source, upflow with radial conduction into rock. Temperature profiles are shown for both water in wellbore and rock at radius one meter. Circles on six-hour curve are from Guyonnet's analytical solution.

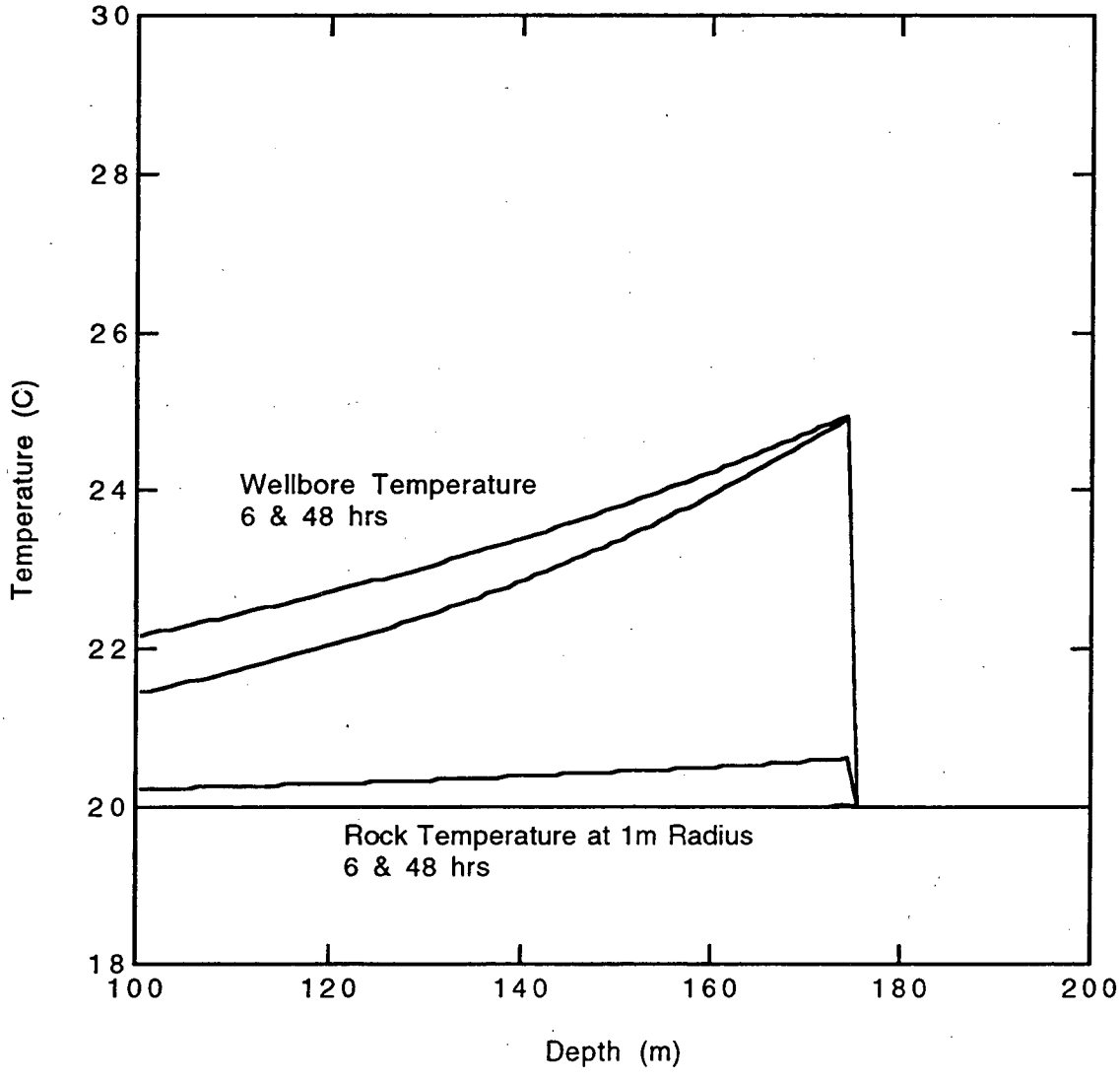


Figure 4. Numerical results for single source, upflow with radial conduction into rock under conditions of constant overall flow. Temperature profiles are shown for both water in wellbore and rock at radius one meter.

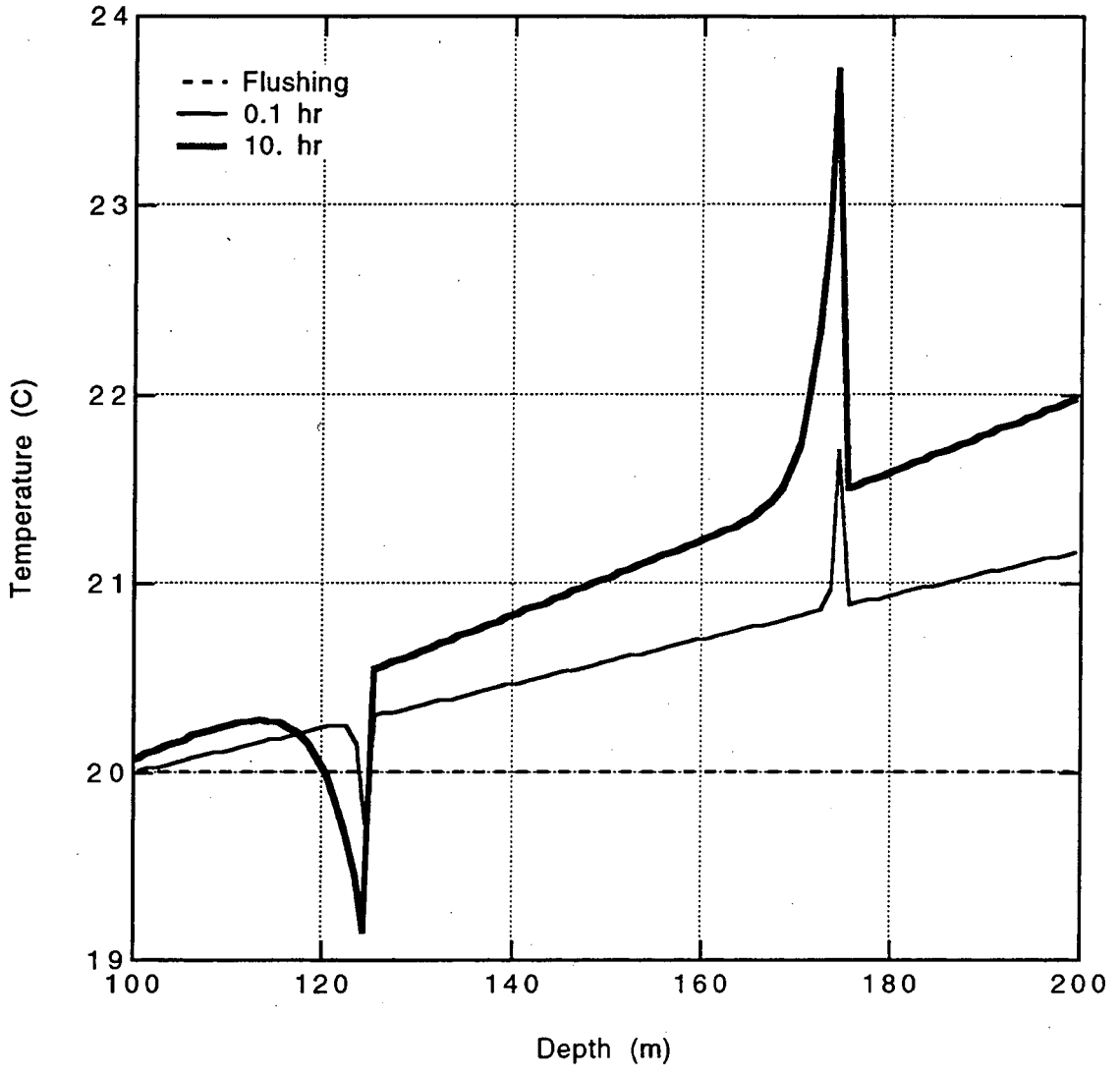


Figure 5. Numerical results for two inflows, one above and one below rock temperature gradient. Temperature profiles are shown for water in wellbore.

Appendix--FORTRAN Source Code

```
c***   borehole heat flow model
c
c***   august 1993
c***   lawrence berkeley laboratory, earth sciences division
c
c***   Only radial heat transfer within rock.
c***   heat only--no mass!
c
c***      for heat, user specifies temp (C) for each inflow,
c***      temp (C) of water in borehole at start, and temp gradient
c***        TA*X + B (C) of rock mass.
c
c***   the following four parameters control array dimensions
c***   in the program.   maxcel is the maximum number of borehole
c***   segments (or rock layers).  maxrng is the maximum number of
c***   concentric rock rings in each layer.   maxinf is the maximum
c***   number of inflows.   maxprt is the maximum number of times
c***   at which temperature profiles are produced.
c
      parameter (maxcel=1000)
      parameter (maxrng=500)
      parameter (maxinf=100)
      parameter (maxprt=25)
c
c***   LEXICON
c
c***   area is cross-sectional area of hole (m**2)
c***   arock is downward thermal gradient for rock (K/m)
c***   brock is rock temperature at depth zero (K)
c***   deal is delta-e-advection-left (J)
c***   dear is delta-e-advection-right (J)
c***   dedl is delta-e-conduction-left (J)
c***   dedr is delta-e-conduction-right (J)
c***   dedrl is delta-e-conduction-rock-left (J)
c***   dedrr is delta-e-conduction-rock-right (J)
c***   dee is delta-e-external (J)
c***   def is delta-e-inflow (J)
c***   dele is delta-e from a rock cell (J)
c***   delt is time step (sec)
c***   delx is borehole segment length, or rock layer thickness (m)
c***   denr is the density for rock: e.g. 2.7e3 kg/m**3
c***   denw is density for water: e.g. 1.e3 kg/m**3
c***   ders is rock diffusion coeff.(J/s*K) [lambda-A/dx]
c***   des in borehole diffusion coeff. (m**3/sec) [lambda-A/c-rho-dx]
c***   diam is borehole diameter (m)
c***   e0 is initial energy (J/m**3)
c***   ecapr is the heat capacity for rock: e.g. 850 J/(kg.K)
c***   ecapw is the heat capacity for water: e.g. 4178.2 J/(kg.K)
c***   econr is the thermal conductivity for rock: e.g. 2.00 J/(s.m.K)
c***   econw is thermal conductivity for water: e.g. 0.607 J/(s.m.K)
c***   efrac is energy in fracture flow into segment (J/m**3)
c***   ein is energy in flow at each fracture (J/m**3)
c***   erb0 initial rock boundary energy (J/m**3)
c***   erock is energy in rock sections (J/m**3)
c***   erockj is energy in a particular rock section (J)
c***   ext is energy in borehole sections (J/m**3)
c***   exti is energy in a particular section (J/m**3)
c***   gte is total energy in system at a given time (J)
c***   gteina is total energy into the system (J)
c***   gteout is total energy out of the system (J)
c***   i is loop index
c***   ie0n is index for initial temperature conditions
c***   iflim is number of fractures
c***   ifmax is maximum number of fractures
c***   ifseg is the segment in which the fracture is found
c***   iin is input unit number
c***   ilim is number of borehole segments (and rock layers)
```

```
c***   ilpt is number of output times
c***   impt is maximum number of print times
c***   iout is general output unit number
c***   iprt is temperature output unit number
c***   ir1 is number of rock rings with thickness diam/8
c***   irlim is number of rock rings
c***   ismax is maximum number of borehole segments (and rock layers)
c***   ist is loop index for time steps
c***   itog is toggle between two sets of C(x,t)
c***   itstep is number of uniform time steps
c***   j is loop index
c***   maxrk is maximum number of rock rings
c***   pi is pi (3.14159...)
c***   pt is print times (sec)
c***   qexti is flow into the bottom (m**3/sec)
c***   qexto is flow out of the top (m**3/sec)
c***   qfrac is fracture flow into each segment (m**3/sec)
c***   qin is flow rate at each fracture (m**3/sec)
c***   qnxt is flow from one segment to the next (up) (m**3/sec)
c***   qtmax is max total flow through a segment (m**3/sec)
c***   qtot is total flow through a segment (m**3/sec)
c***   r1 is inside radius of rock ring (m)
c***   r2 is outside radius of rock ring (m)
c***   rad1 is radial extent of thin rock rings (m)
c***   rinc is thickness of rock ring (m)
c***   rockr is radial distance to rock ring midpoint (m)
c***   rockv is the volume of a rock ring segment (m**3)
c***   t is time at end of time step (sec)
c***   t0 is the starting time for fracture flow at temperature (sec)
c***   te0 is initial energy in system, borehole and rock (J)
c***   tend is ending time (sec)
c***   temp is a temporary variable
c***   ter0 is initial energy in rock (J)
c***   totin is total flow into a cell (m**3/sec)
c***   totout is total flow out of a cell (m**3/sec)
c***   tp is time at beginning of time step (sec)
c***   tstart is starting time (sec)
c***   vol is uniform borehole segment volume (m**3)
c***   x is depth (increasing downward) (m)
c***   xbot is bottom of analysis region (m)
c***   xin is position of fracture (m)
c***   xs is segment midpoint (m)
c***   xtop is top of analysis region (m)
c
      dimension erb0(maxcel)
      dimension pt(maxprt)
      dimension ext(maxcel,2),efrac(maxcel),
     .          qfrac(maxcel),qnxt(maxcel),
     .          qtot(maxcel),xs(maxcel),
     .          erock(maxcel,maxrng,2),rockr(maxrng),
     .          rockv(maxrng),ders(maxrng)
      dimension qin(maxinf),xin(maxinf),ifseg(maxinf),t0(maxinf),
     .          ein(maxinf)
c
      data iout/6/
c
c***   input and initialization
c
      ismax  = maxcel
      ifmax  = maxinf
      maxrk  = maxrng
      impt   = maxprt
c
      open ( unit=5, file='BORE.DAT',status='OLD',
     .       form='FORMATTED',access='SEQUENTIAL' )
      open ( unit=6, file='BORE.MSG', status='UNKNOWN',
     .       form = 'FORMATTED', access = 'SEQUENTIAL' )
      open ( unit=7, file = 'BORE.OUT', status = 'UNKNOWN',
     .       form = 'FORMATTED', access = 'SEQUENTIAL' )
c
      call rdprob(delt,denr,denw,ders,des,
     .            ecapr,ecapw,ein,erb0,erock,ext,
     .            iflim,ifmax,ifseg,ilim,ilpt,impt,
     .            iout,irlim,ismax,itstep,maxrk,pt,
     .            qexto,qfrac,qin,qnxt,qtot,rockr,rockv,
```

```
     .           t0,te0,tend,ter0,tstart,vol,xin,xs,ir1)
      call tsteps(delt,denr,denw,ders,des,
     .           ecapr,ecapw,efrac,ein,erb0,erock,ext,
     .           iflim,ifseg,ilim,ilpt,
     .           iout,irlim,ismax,itstep,maxrk,pt,
     .           qexto,qfrac,qin,qnxt,rockv,rockr,
     .           t0,te0,tend,ter0,tstart,vol,xs,ir1)
      write(iout,'(//'' *** PROBLEM FINISHED ***'')')
      stop
c
      end
c
c***   subroutine rdprob
c***   read problem description
c
      subroutine rdprob(delt,denr,denw,ders,des,
     .                  ecapr,ecapw,ein,erb0,erock,ext,
     .                  iflim,ifmax,ifseg,ilim,ilpt,impt,
     .                  iout,irlim,ismax,itstep,maxrk,pt,
     .                  qexto,qfrac,qin,qnxt,qtot,rockr,rockv,
     .                  t0,te0,tend,ter0,tstart,vol,xin,xs,ir1)
c
      dimension qfrac(ismax),qnxt(ismax),qtot(ismax),xs(ismax),
     .          erb0(ismax)
      dimension rockr(maxrk),rockv(maxrk),ders(maxrk)
      dimension qin(ifmax),xin(ifmax),ifseg(ifmax),t0(ifmax),
     .          ein(ifmax)
      dimension pt(impt)
      dimension ext(ismax,2),erock(ismax,maxrk,2)
c
      data pi/3.141592654/, iin/5/
c
c***   read basic borehole geometry
c***   convert diameter in cm to m
c
      read(iin,*)xtop,xbot,delx,diam
      diam=diam/100.
c
c***   check geometry
c
      if (xbot.lt.xtop) then
         write(iout,*)'Aborting--Top of region below bottom'
         stop 101
      endif
      ilim=(xbot-xtop)/delx
      if (ilim.lt.1) then
         write(iout,*)'Aborting--delta-x is too big'
         stop 102
      endif
      if (ilim.gt.ismax) then
         write(iout,*)'Aborting--Max segments is ',ismax
         stop 103
      endif
      temp=xtop
      xtop=xbot-(ilim*delx)
c
c***   if delta-x does not fit over the interval within 1% of
c***   a delta-x, then make a note
c
      write(iout,*)'Borehole segments:',ilim
      write(iout,*)'Segment length (m): ',delx
      if (abs(xtop-temp).gt.(0.01*delx))
     .   write(iout,*)'NOTE--Top of region has been moved to (m):',
     .    xtop
c
c***   calculate cross-sectional area and volume of a segment
c
      area=pi*(diam*diam/4.)
      vol=area*delx
      write(iout,*)' Each region has diameter         (m)   ',diam
      write(iout,*)'             cross-sectional area (m**2)',area
      write(iout,*)'         and volume               (m**3)',vol
c
c***   compute segment midpoints
```

```
c***   note--sign reversed for graphing
c
      do i=1,ilim
         xs(i)=xbot+(0.5-i)*delx
      enddo
c
c***   read rock properties
c
      read(iin,*)rad1,ecapr,econr,denr
c
c***   ir1 is number of small rings
c***   ir2 is number of big rings
c***   big rings extend to 9 times rad1
c***   big rings increase geometrically by 1.2
c
      irlim  = 1 + ( (rad1-(diam/2.)) / (diam/8.) ) +
     .       ( alog10(1.2+((1.2-1)*(9.*rad1)/(diam/8.))) /
     .         alog10(1.2) )
      if (irlim.gt.maxrk) then
         write(iout,'(''*** error: max rock rings: '',i5)')maxrk
         stop 104
      endif
c
c***   compute radii, volumes and (A/del-x) for rock sections
c
      write(iout,*)' '
      write(iout,*)'Rock ring, min & max radius (m), vol (m**3)'
      r1   = diam/2.
      rinc = diam/8.
      ir1  = 1
      do i=1,irlim
         if (r1.gt.rad1) rinc = rinc * 1.2
         if (r1.le.rad1) ir1 = i
         r2 = r1 + rinc
         rockr(i) = 0.5*(r1+r2)
         rockv(i) = delx*pi*(r2*r2 - r1*r1)
         if (i.eq.1) then
            ders(i)  = ((2.*pi*r1)*delx)  / (rockr(i)-(diam/2.))
         else
            ders(i)  = ((2.*pi*r1)*delx)  / (rockr(i)-rockr(i-1))
         endif
         write(iout,'(i3,2f10.4,2f16.5)')i,r1,r2,rockv(i)
         r1 = r2
      enddo
c
c***   top and bottom flows
c***   convert from l/min to m**3/sec
c
      read(iin,*)qexti,qexto
      qexti = qexti / 60000.
      qexto = qexto / 60000.
c
c***   read fracture data
c
      read(iin,*)iflim,ecapw,econw,denw
      if (iflim.gt.ifmax) then
         write(iout,*)'Aborting--max number of fractures is ',ifmax
         stop 105
      endif
c
      do i=1,iflim
         read(iin,*)xin(i),qin(i),ein(i),t0(i)
c
c***   locate fracture on segments
c
         if ((xin(i).gt.xbot) .or. (xin(i).lt.xtop)) then
            write(iout,*)'Aborting--fracture outside of region: ',
     .         xin(i)
            stop 106
         endif
         ifseg(i)=iconvx(xtop,xbot,delx,xin(i))
c
c***   convert qin from l/min to m**3/sec
c
         qin(i) = qin(i) / 60000.
```

```
c
c***   convert t0 to seconds
c
         t0(i)=t0(i)*3600.
c
c***   convert fracture inflow temperature (C) to J/m**3
c
         ein(i) = (ein(i)+273.15) * ecapw * denw
       enddo
c
c***   initialize segment flows
c
       do i=1,ilim
          qfrac(i)=0.
       enddo
c
c***   compute flow into segments from fractures
c***   it is possible to have more than one fracture in a single
c***   segment
c
       do i=1,iflim
          qfrac(ifseg(i))=qfrac(ifseg(i))+qin(i)
       enddo
c
c***   compute flow up to next segment and total flow
c
       temp = qexto
       do i=1,ilim
          if (i.eq.1) then
             qtot(i)=qexti+qfrac(i)
             qnxt(i)=qtot(i)
          else
             if (qfrac(i).eq.0.) then
                qtot(i) = abs(qnxt(i-1))
                qnxt(i) = qnxt(i-1)
             else
                qnxt(i) = qnxt(i-1)+qfrac(i)
                qtot(i) = (abs(qnxt(i))+abs(qnxt(i-1))+
                           abs(qfrac(i)))/2.
             endif
          endif
       enddo
c
c***   adjust input value of outflow at top to match total flow
c
       if (qnxt(ilim-1).ge.0.) then
          qexto=qtot(ilim)
       else
          qexto=-qtot(ilim)
       endif
       qnxt(ilim) = 0.
       if (abs(temp-qexto).gt. (0.01*temp)) then
          write(iout,*)'Outflow at top changed by ',abs(temp-qexto)
          write(iout,*)'                         to ',qexto
       endif
c
c*** check each cell one last time
c
       do i=1,ilim
          totin  = 0.
          totout = 0.
          if (i.eq.1)    totin  = qexti
          if (i.eq.ilim) totout = qexto
          if (qfrac(i).gt.0.) totin  = totin  + qfrac(i)
          if (qfrac(i).lt.0.) totout = totout - qfrac(i)
          if (i.gt.1) then
             if (qnxt(i-1).gt.0.) totin  = totin + qnxt(i-1)
             if (qnxt(i-1).lt.0.) totout = totout- qnxt(i-1)
          endif
          if (i.lt.ilim) then
             if (qnxt(i).lt.0.) totin  = totin - qnxt(i)
             if (qnxt(i).gt.0.) totout = totout+ qnxt(i)
          endif
          if (abs(totin-totout)/(0.5*(totin+totout))
              .gt.0.01) then
```

```
            write(iout,*)'*** flow continuity error--cell ',i,xs(i)
            write(iout,*)'  Total in =',totin
            write(iout,*)'  Total out=',totout
            write(iout,*)'   qexti, qexto=',qexti,qexto
            write(iout,*)'   qfrac       =',qfrac(i)
            if (i.gt.1) write(iout,*)'qnxt(i-1) = ',qnxt(i-1)
            if (i.lt.ilim) write(iout,*)'qnxt(i) = ',qnxt(i)
            stop 107
         endif
         if (qtot(i).ne.abs(totin)) then
            write(iout,*)'*** correcting total flow--cell ',i
            write(iout,*)'   from ',qtot(i),' to ',abs(totin)
            qtot(i) = abs(totin)
         endif
      enddo
c
c***   compute maximum time step and Reynold's number
c
      qtmax = abs(qtot(1))
      do i=2,ilim
         if (abs(qtot(i)).gt.qtmax) qtmax=abs(qtot(i))
      enddo
c
c***   maximum time step is time minimum time to flush one-half
c***   of a segment.  note that this time may still be too large
c***   if dispersion is significant
c
      write(iout,*)'Maximum time step (min) is         ',
     .     ((0.5*vol/qtmax)/60.)
      write(iout,*)'    (based on advection only)'
      write(iout,*)'Maximum linear velocity (m/sec) is ',
     .     (qtmax/area)
      write(iout,*)'Maximum Reynold''s number is       ',
     .     ((qtmax/area)*diam/0.3e-6)
c
c***   read in time parameters
c
      read(iin,*)tstart,tend,delt,ilpt
      tstart = tstart*3600.
      tend   = tend  *3600.
      delt   = delt  *  60.
c
      if (tend.lt.tstart) then
         write(iout,*)'End time before start time -- aborting'
         stop 108
      endif
      if (ilpt.gt.impt) then
         write(iout,*)'Aborting--max print times is ',impt
         stop 109
      endif
c
c***   read in print-time parameters
c
      read(iin,*)(pt(i),i=1,ilpt)
      do i=1,ilpt
         pt(i)=pt(i)*3600.
      enddo
c
c***   check for maximum delta-t
c
      if (delt.gt.(0.5*vol/qtmax)) then
         delt = 0.5*vol/qtmax
         write(iout,*)'NOTE--time step set to maximum (min):',
     .     (delt/60.)
      endif
c
c***   compute number of time steps
c
      temp  = tend
      itstep = (tend-tstart)/delt
      tend   = tstart+itstep*delt
      if ((temp-tend).lt.0.) itstep=itstep+1
      if (itstep.gt.1000000) then
         write(iout,*)'Aborting---max time steps is 1,000,000'
         stop 110
```

```
      endif
c
c***   adjust ending time if needed
c
      if (abs(tend-temp).gt.(0.01*delt))
     .   write(iout,*)'NOTE--Ending time changed to (hrs) ',
     .    (tend/3600.)
      write(iout,*)'Number of time steps:              ',itstep
      write(iout,*)'Time step (min):                   ',
     .      (delt/60.)
      write(iout,*)'Problem size (segments*time steps):',
     .      (itstep*ilim*irlim)
c
c***   check fracture flow times to make sure they fit into problem
c
      do i=1,iflim
         if ((t0(i).lt.tstart) .or. (t0(i).gt.tend)) then
            write(iout,*)'NOTE--fracture ',i
            write(iout,*)'   t0 outside of problem time limits: ',
     .         t0(i)/3600.
         endif
      enddo
c
c***   set up initial temperature conditions in borehole
c
      do i=1,ilim
         ext(i,1)=0.
      enddo
      te0 = 0.
      e0  = 0.
c
      read(iin,*)ie0n
c
c***   if ie0n gt 0, specifying initial values for each cell
c
      if (ie0n .gt. 0) then
         do i=1,ie0n
            read(iin,*)x,e0
            e0 = (e0+273.15)*ecapw*denw
            if ((x.gt.xbot) .or. (x.lt.xtop)) then
               write(iout,*)'NOTE--invalid initial position',x
            else
               if (ext(iconvx(xtop,xbot,delx,x),1).eq.0.) then
                  ext(iconvx(xtop,xbot,delx,x),1) = e0
                  te0 = te0 + vol*ext(iconvx(xtop,xbot,delx,x),1)
               else
                  write(iout,*)'NOTE--multiple initial temp at ',x
                  write(iout,*)'      second value ignored'
               endif
            endif
         enddo
         e0 = 0.
      else
c
c***   if ie0n lt 0, read constant temperature for borehole
c
         if (ie0n.lt.0) then
            read(iin,*)e0
            e0 = (e0+273.15)*ecapw*denw
            do i=1,ilim
               ext(i,1) = e0
            enddo
            te0 = (ext(1,1)*vol)*ilim
         endif
      endif
      write(iout,*)'Initial heat in borehole (J):          ',te0
c
c***   read rock temperature coefficients
c***    t = arock*x + brock
c
      read(iin,*)arock,brock
      brock = brock + 273.15
      ter0   = 0.
      do i=1,ilim
         do j=1,irlim
```

```
            erock(i,j,1) = (arock*xs(i) + brock)*ecapr*denr
         enddo
         erb0(i) = erock(i,irlim,1)
      enddo
      do j=1,irlim
         ter0 = ter0 +
     .          ilim*0.5*(erock(1,j,1)+erock(ilim,j,1))*rockv(j)
         ders(j) = ders(j) * econr
      enddo
      write(iout,*)'Initial heat in rock (J):              ',ter0
      te0 = te0 + ter0
      write(iout,*)'Initial heat in system (J):            ',te0
      write(iout,*)
      write(iout,*)
c
      des = (econw * area ) / (ecapw * denw * delx)
c
      return
      end
c
c***   subroutine cprt
c***   print out temperature arrays
c***   this subroutine is executed whenever temperature arrays
c***   are requested during the time steps, or at the end of the
c***   problem
c
      subroutine cprt (t,ilim,ext,denw,ecapw,xs,iout,maxrk,
     .                 itog,ismax,ir1,erock,irlim,
     .                 ecapr,denr,rockr)
c
      dimension ext(ismax,2),xs(ilim)
      dimension erock(ismax,maxrk,2),rockr(irlim)
      data iprt/7/
c
c***   output time in hours and temperature with ascending
c***   values (note negative increment in loop)
c***   output temperatures for wellbore, limit of small rock
c***      rings, outside limit of rock
c
      write(iprt,*)
      write(iprt,*)'Time (hr) =',(t/3600.)
      write(iprt,*)' r1   (m) =',rockr(ir1)
      write(iprt,*)' rmax (m) = ',rockr(irlim)
      do i=ilim,1,-1
         write(iprt,'(f10.3,3f10.4)')xs(i),
     .   (ext  (i,      itog) / (denw*ecapw) - 273.15),
     .   (erock(i,ir1,  itog) / (denr*ecapr) - 273.15),
     .   (erock(i,irlim,itog) / (denr*ecapr) - 273.15)
      enddo
      write(iout,*)'Printing profile at time (hrs)',(t/3600.)
      return
c
      end
c
c***   function iconvx
c***   convert x position to segment index
c***   segment 1 begins at xbot
c***   segment ilim ends at xtop
c***   this function is used outisde of time steps
c
      function iconvx(xtop,xbot,delx,x)
c
      if (x.eq.xtop) then
         iconvx = (xbot-x) / delx
      else
         iconvx = (xbot-x) / delx+1.
      endif
      return
      end
c
      subroutine tsteps(delt,denr,denw,ders,des,
     .                  ecapr,ecapw,efrac,ein,erb0,erock,ext,
     .                  iflim,ifseg,ilim,ilpt,
```

```
     .                 iout,irlim,ismax,itstep,maxrk,pt,
     .                 qexto,qfrac,qin,qnxt,rockv,rockr,
     .                 t0,te0,tend,ter0,tstart,vol,xs,ir1)
c
      dimension efrac(ilim),ifseg(iflim),erock(ismax,maxrk,2),
     .          ext(ismax,2),pt(ilpt),qfrac(ilim),ein(iflim),
     .          t0(iflim),qnxt(ilim),ders(irlim),erb0(ilim),
     .          qin(iflim),rockv(irlim),rockr(irlim)
c
      data itog/1/
c
c***   output initial conditions
c
      call cprt (tstart,ilim,ext,denw,ecapw,xs,iout,maxrk,
     .           itog,ismax,ir1,erock,irlim,
     .           ecapr,denr,rockr)
c
c***   loop over each time step
c
      do ist=1,itstep
c
c***   compute interval time limits
c
        t  = tstart+ist*delt
        tp = t-delt
c
c***   reset values first
c
        do i=1,iflim
           efrac(ifseg(i))=0.
        enddo
c
c***   compute energy inflow (J/m**3) during time step
c***   more than one fracture may be in a segment
c
        do i=1,iflim
c
           iseg = ifseg(i)
           if (qin(i).gt.0.) then
c
              if (t.le.t0(i)) then
c
c***   before t0, use wellbore fluid characteristics
c
                 efrac(iseg) = efrac(iseg) +
     .                         ext(iseg,itog)*qin(i)/qfrac(iseg)
              else
                 if (tp.ge.t0(i)) then
c
c***   after t0, use inflow fluid characteristics
c
                    efrac(iseg) = efrac(iseg) +
     .                            ein(i)*qin(i)/qfrac(iseg)
                 else
c
c***   crossing t0 in time step, use a mix
c
                    efrac(iseg) = efrac(iseg) +
     .                              ((t0(i)-tp)/delt)*
     .                                ext(iseg,itog)*
     .                                qin(i)/qfrac(iseg)  +
     .                              (1.-((t0(i)-tp)/delt))*
     .                                ein(i)*qin(i)/qfrac(iseg)
                 endif
              endif
           else
              efrac(iseg) = efrac(iseg) +
     .                      ext(iseg,itog)*qin(i)/qfrac(iseg)
           endif
        enddo
c
c***   calculate energy transfer
c
      gte  = 0.
c
```

```
c***   loop over each segment
c
       do i=1,ilim
          exti = ext(i,itog)
          if (efrac(i).ne.0.) then
            def=efrac(i)*qfrac(i)
          else
             def=0.
          endif
c
c***   generalized outflow disabled
c***   outflow considered only at top segment
c
          if (i.gt.1) then
c
c***   independent calculation of energy transfer with left side
c***   disabled--using complement of right side values from below
c
             deal=-dear
             dedl=-dedr
          else
             deal=0.
             dedl=0.
          endif
          if (i.lt.ilim) then
c
c***   note:  full upstream weighting is used.  without
c***          upstream weighting, these equations would
c***          use (exti+ext(i+1,itog))/2. in place of
c***          exti or ext(i+1,itog)--that is, the
c***          temperature at the interface would be
c***          the average instead of the upstream value
c
             if (qnxt(i).ge.0.) then
                dear=-exti*qnxt(i)
             else
                dear=-ext(i+1,itog)*qnxt(i)
             endif
             if (qfrac(i).ge.0. .and. qfrac(i+1).ge.0.) then
                dedr=(ext(i+1,itog)-exti)*des
             else
                dedr = 0.
             endif
             dee = 0.
          else
             dear=0.
             dedr=0.
c
c***   outflow considered only at top
c
             dee=-exti*qexto
          endif
c
c***   do heat conduction within rock and between rock and
c***   borehole
c
          do j=1,irlim
             erockj = erock(i,j,itog) * rockv(j)
c
c***   conduction on "borehole or inward side"
c
             if (j.eq.1) then
c
c***   rock layer adjacent to borehole
c
                dedrl = ((exti/(denw*ecapw))-
                         (erock(i,j,itog)/(denr*ecapr))) * ders(1)
                der   = -dedrl
             else
                dedrl = -dedrr
             endif
c
c***   conduction on "outward side"
c
             if (j.ne.irlim) then
```

```
               dedrr = ((erock(i,(j+1),itog)-
     .                   erock(i,j    ,itog)) / (denr*ecapr) ) *
     .                  ders(j+1)
            else
c
c***   nonconducting boundary at outside of rock layers
c
               dedrr = 0.
            endif
c
            dele = (dedrl + dedrr)*delt
            if (-dele.gt.(0.5*erockj)) then
               write(iout,*)'Too much energy out of rock cell'
               write(iout,*)'  try reducing time step to (min):',
     .              (((-0.25*erockj)/(dedrl+dedrr))/60.)
               stop 180
            endif
            erock(i,j,3-itog) = (erockj + dele) / rockv(j)
            gte = gte+(erockj+dele)
         enddo
c
c***   check that outside rock segment has not changed more than
c***     one degree
c
         if (abs(erock(i,irlim,3-itog)-erb0(i)).gt.
     .        denr*ecapr) then
            write(*,*)'Aborting--rock volume too small,'
            write(*,*)'   Increase input rock radius.'
            write(*,*)'   Change of more than one degree in'
            write(*,*)'   outside ring at cell ',i
            stop 200
         endif
c
c***   compute new energy in segment
c
         ext(i,3-itog) = exti +
     .                   ((def+dee+deal+dedl+dear+dedr+der)
     .                     *delt)/vol
c
c***   check for conservation of energy
c
         gteout = gteout - (dee*delt)
         if (def.gt.0.) then
            gteina = gteina + (def*delt)
         else
            gteout = gteout - (def*delt)
         endif
         gte = gte + ext(i,3-itog)*vol
      enddo
c
c***   end of loop over segments
c
c***   check for conservation of energy
c
      if (gteina.ne.0.) then
         temp=(100.* (gte-(te0+gteina-gteout)))/(te0+gteina)
         if (abs(temp).gt.0.1) then
            write(iout,*)'Energy error (%)       = ',temp
            write(iout,*)'   Initial energy (MJ) = ',(te0/1.e6)
            write(iout,*)'   Added (MJ)          = ',(gteina/1.e6)
            write(iout,*)'    Out  (MJ)          = ',(gteout/1.e6)
            write(iout,*)'  Expected (MJ)        = ',
     .           ((te0+gteina-gteout)/1.e6)
            write(iout,*)'  Observed (MJ)        = ',(gte/1.e6)
            stop 111
         endif
      endif
c
c***   toggle row pointer
c
      itog=3-itog
c
c***   check for output requests
c
         do j=1,ilpt
```

```
            if (t.ge.pt(j)) then
               call cprt (t,ilim,ext,denw,ecapw,xs,iout,maxrk,
     .                    itog,ismax,ir1,erock,irlim,
     .                    ecapr,denr,rockr)
               pt(j)  = 2.*tend
            endif
         enddo
c
c***   end of loop over time steps
c
      enddo
c
c***   check for final print
c
      temp = 0.
      do j=1,ilpt
         if (pt(j).lt.(2.*tend)) temp = 1.
      enddo
      if (temp .ne. 0.)
     .    call cprt (t,ilim,ext,denw,ecapw,xs,iout,maxrk,
     .               itog,ismax,ir1,erock,irlim,
     .               ecapr,denr,rockr)
c
c***   final information messages about heat
c
      write(iout,*)
      write(iout,*)
      temp = 0.
      do i=1,ilim
         temp = temp + ext(i,itog)
      enddo
      temp = temp * vol
      write(iout,*)'Final heat in borehole (J):           ',
     .      temp
      write(iout,*)'Change in heat in borehole (J):       ',
     .      (temp-(te0-ter0))
      write(iout,*)'Final heat in rock (J):               ',
     .      (gte-temp)
      write(iout,*)'Change in heat in rock (J):           ',
     .      ((gte-temp)-ter0)
      write(iout,*)'Final heat in whole system (J):       ',
     .      gte
      write(iout,*)'Change in heat in whole system (J):   ',
     .      (gte-te0)
      write(iout,*)'Heat added to system (J):             ',
     .      gteina
      write(iout,*)'Heat removed from system (J):         ',
     .      gteout
      return
c
      end
```

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
TECHNICAL INFORMATION DEPARTMENT
BERKELEY, CALIFORNIA 94720